

Algorithm for Wave- Particle Resonances in Fluid Codes – Final Report

N. Mattor

March 24, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Work performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
<http://apollo.osti.gov/bridge/>

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
<http://www.ntis.gov/>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

Algorithm for Wave-Particle Resonances in Fluid Codes

- Final Report -

Nathan Mattor

Lawrence Livermore National Laboratory
Livermore, California 94550

March 3, 2000

Abstract

We review the work performed under LDRD ER grant 98-ERD-099. The goal of this work is to write a subroutine for a fluid turbulence code that allows it to incorporate wave-particle resonances (WPR). WPR historically have required a kinetic code, with extra dimensions needed to evolve the phase space distribution function, $f(x, v, t)$. The main results accomplished under this grant have been:

- (1) Derivation of a nonlinear closure term for 1D electrostatic collisionless fluid
- (2) Writing of a 1D electrostatic fluid code, "eslf," with a subroutine to calculate the aforementioned closure term
- (3) derivation of several methods to calculate the closure term, including Eulerian, Euler-local, fully local, linearized, and linearized zero-phase-velocity, and implementation of these in eslf,
- (4) Successful modeling of the Landau damping of an arbitrary Langmuir wave,
- (5) Successful description of a kinetic two-stream instability up to the point of the first bounce, and
- (6) a spin-off project which uses a mathematical technique developed for the closure, known as the Phase Velocity Transform (PVT) to decompose turbulent fluctuations.

I. Introduction

In 1946, Landau discovered an effect ("Landau damping") that occurs in collisionless fluids and plasmas, wherein particles resonate with waves and cause growth or damping of the wave. Since then, numerous studies have extended this effect to many different regimes. One belief that persisted to recent times is that theories of Landau Damping require retaining the entire phase-space distribution, $f(x, v, t)$, which is generally quite complicated. However, starting with Ott and Sudan (1969), various closures have been able to include assorted aspects of Landau damping in fluid equations, with a "collisionless closure term" (CCT) that completes the collisionless fluid equations much as a viscosity does in the collisional limit. The most recent stage of this progression was a finding by myself, in 1997, of a method that includes in the CCT nearly all of the resonance effects present in the fluid equations (in principle). Fluid equations have half the dimensions of kinetic, so this promises to simplify greatly understanding of many systems.

What remained was a demonstration that this new fluid "CCT" technique offers an actual improvement over current kinetic techniques of predicting collisionless fluids. This has been the goal of the current LDRD project. We choose the simplest system known to experience a variety of wave-particle resonance effects, the 1D electrostatic fluid. An existing kinetic particle code, es1 [Birdsall and Langdon, 1985], solves this via the conventional kinetic equations, and provides a benchmark with which to compare. The ultimate goal is to have a fluid code, eslf, that can do everything es1 can but with far less computational time. The intended "blueprint" for the project is as follows:

1. Derive the CCT for a 1D electrostatic fluid.
2. Write a 1D electrostatic fluid code that allows inclusion of the above term.
3. Consider various known wave-particle resonance effects, such as linear landau damping, trapping, plasma echoes, various kinds of scattering, cavitons and so forth, and try to get eslf to exhibit these.
4. Experiment with various forms of calculating the closure term to see which is simpler, faster, or more able to obtain a variety of the above resonance effects.

None of these are entirely straightforward: multiple choices exist for each step, and it is not *a priori* clear which is best. In the following, we describe how this work has progressed toward these goals. The work is not yet complete, and but there has been considerable progress.

In addition to the main goals above, there have been two spin-off results, which are (1) generalization of the closure to general Hamiltonian system, (which encompasses pretty much any collisionless system in existence), and (2) using a new piece of mathematics developed in the theory (the Phase Velocity Transform) as a tool to analyze spatio-temporal fluctuations. These are described below.

The remainder of this work is organized as follows.

Section II gives the collisionless 1D electrostatic kinetic equation, and the corresponding kinetic equation with the closure term used to copy it. Section III discusses various methods derived for calculating the closure term. Section IV discusses the two physical scenarios we have addressed thus far, the O'Neil Problem and the Two-Stream Instability. Section V discusses the generalization of the CCT derived for a general collisionless Hamiltonian system. Section VI discusses a spin-off application of the work here, which is using the Phase Velocity Transform, an integral transform developed to derive the CCT, to analyze spatio-temporal fluctuation data.

II. Model, Equations, and Collisionless Closure Term

The system we model is a 1D periodic domain of length L , containing an arbitrary number of particle species α , each with its own charge q_α , mass m_α , and phase distribution function $f_\alpha(x, v, t)$. Through Poisson's equation these species generate an electric field, $E = -\nabla\phi$. This system evolves by a Vlasov equation for each species α ,

$$\partial_t f_\alpha + v \partial_x f_\alpha + (q_\alpha/m_\alpha) E \partial_v f_\alpha = 0, \quad (2.1)$$

and Poisson's equation

$$\partial_x^2 \phi = -4\pi \sum_\alpha q_\alpha n_\alpha. \quad (2.2)$$

Fluid moments of Eq. (1) can be taken, giving the familiar fluid equations,

$$\partial_t n_\alpha + \partial_x (nV)_\alpha = 0 \quad (2.3)$$

$$\partial_t nV_\alpha + \partial_x (nVV)_\alpha = (q_\alpha/m_\alpha) E n_\alpha - \partial_x \nabla P_\alpha. \quad (2.4)$$

As usual, for the moment hierarchy to be closed requires the pressure P to be calculated. Refs. (MattorII-MattorIV) explain the techniques to do this for Eq. (2.1), and we do not

repeat the details here because it is not the main focus of this paper. There are numerous possible forms of P , but the one we will focus on here is:

$$P \simeq -mnV^2 + T_0\tilde{n} + \int_{-\infty}^{\infty} dv_p \sum_{s=\pm} v_{th} B_s(w_p) \left[\tilde{n} \tilde{V}^{\dagger s} - (v_{pp} - v_p) \tilde{n}^{\dagger s} \right]. \quad (2.5)$$

Defining the terms in Eq. (2.5) is fairly cumbersome, and proceeds as follows. First, $B_{\pm}(x)$ is a special function,

$$B_{\pm}(x) = -\frac{Z_{\pm}^{[2]}(x/\sqrt{2})}{Z_{\pm}^{[1]}(x/\sqrt{2})},$$

where $Z_{\pm}^{[n]}$ is the n^{th} derivative of the plasma dispersion function, defined as $Z_{\pm}^{[n]}(x/\sqrt{2}) = \partial_x^n Z_{\pm}(x/\sqrt{2})$. The argument of B_s is w_p , the relative phase velocity normalized to the thermal velocity,

$$w_p \equiv [v_{pp}(x, v_p, t; t_0) - V_0] / v_{th},$$

where $v_{p,p}$ is the accelerated phase velocity, defined as the solution to

$$\begin{aligned} d_{t'} x_p(q, v_p, t; t') &= v_p p(x, v_p, t; t'), \\ d_{t'} v_{pp}(q, v_p, t; t') &= E(x_p, t'), \end{aligned}$$

with the boundary conditions

$$\begin{aligned} x_p(x, v_p, t; t) &= x, \\ v_{pp}(q, v_p, t; t) &= v_p. \end{aligned}$$

The term \tilde{n}^{\dagger} (or $\tilde{n} \tilde{V}^{\dagger}$) is the phase velocity transform of $n(x, t)$, defined as

$$\tilde{n}^{\dagger}(x, v_p, t) \equiv -\frac{1}{2\pi} \mathcal{H}_x \int_{t_0}^{\infty} dt' \int dx' \delta[x' - x_p(x, v_p, t; t')] \partial_{x'} n(x', t'),$$

\mathcal{H}_x is a Hilbert transform in x

$$\mathcal{H}_x g(x) \equiv \frac{P}{\pi} \int_{-\infty}^{\infty} dx' \frac{g(x')}{x' - x}.$$

The inverse PVT is

$$\tilde{n}(x, t) \simeq \int_{-\infty}^{\infty} dv_p \tilde{n}^{\dagger}(x, t).$$

Finally, the equilibrium terms n_0 , V_0 , and T_0 , are the (constant) density, velocity, and temperature of the initially specified Maxwellian for species α , and $v_{th} = T_0^{1/2}$,

$$F_{M0}(x, v) = \frac{n_0(x)}{\sqrt{2\pi T_0}} \exp \left\{ -\frac{(v - V_0)^2}{2T_0(x)} \right\}.$$

Details of the derivation of Eq. (5) can be found in *MattorII*.

III. Methods to compute the CCT

The computation of the P given by Eq. (5) is clearly a challenge. Part of the work for the LDRD has been formulating several schemes for this. This section describes these schemes, and their advantages and disadvantages.

The two challenging terms in Eq. (5) are the accelerated phase velocity, $v_{p,p}(x, v_p, t; t_0)$, and the transformed velocity and density, $\tilde{n}^\dagger(x, v_p, t)$ and $\tilde{n}\tilde{V}^\dagger(x, v_p, t)$. Possible methods include Eulerian and Lagrangian schemes in either x or v_p , local approximations, linearization, and using Fourier transforms. Over the last year, I have developed and tested several schemes, with some success and some failures. Below is a list of the main ones.

A. Full Eulerian calculation

The full Eulerian calculation is a “complete” method, in that it computes Eq. (5) to arbitrary numerical accuracy. The basic idea is to derive Eulerian differential equations for \tilde{n}^\dagger and $\tilde{n}\tilde{V}^\dagger$, evolve them numerically on an Eulerian grid, and perform the integral in Eq. (5) based on knowledge of the full v_p spectrum.

A small amount of manipulation of the definitions of \tilde{n}^\dagger and $\tilde{n}\tilde{V}^\dagger$ produces the equations

$$\{\partial_t + v_p \partial_x + E \partial_{v_p}\} \tilde{n}^\dagger = 0, \quad (3.1)$$

$$\{\partial_t + v_p \partial_x + E \partial_{v_p}\} \tilde{n}\tilde{V}^\dagger = E \tilde{n}^\dagger, \quad (3.2)$$

with the initial conditions

$$\tilde{n}^{\dagger\pm}(\zeta, t_0) = \frac{-n_0}{2\pi\sqrt{2T_0}} Z_\pm, \quad (3.4)$$

$$\tilde{n}\tilde{V}^{\dagger\pm}(\zeta, t_0) = \frac{-n_0}{2\pi\sqrt{2T_0}} (-T_0 \partial_{v_p} + V_0) Z_\pm, \quad (3.5)$$

where the argument of Z_\pm is $[(v_p - V_0)/\sqrt{2}v_{th}]$. Eqs. (3.1-3.2) are evolved, and then $\tilde{n}^\dagger = \mathcal{H}_x \tilde{n}^\dagger$ and $\tilde{n}\tilde{V}^\dagger = \mathcal{H}_x \tilde{n}\tilde{V}^\dagger$ are used to get the fully transformed fields. Details of the derivation are given in Sec. 3 of MattorIV.

Eqs. (3.1-3.2) can be solved straightforwardly by creating a grid in x and v_p space, and stepping \tilde{n}^\dagger and $\tilde{n}\tilde{V}^\dagger$ by the usual finite difference scheme. The approach thus resembles the standard “Vlasov method” of solving Eq. (1) for $f(x, v, t)$, which uses a grid in x, v space. There are no great time savings, but the hope was to create a “benchmark” solution to compare with later approximate methods.

Unfortunately, it has proven difficult to find a good boundary condition in the v_p dimension for the full Eulerian calculation. A kinetic Vlasov code has no problem in the v dimension, because the *kinetic* distribution function is basically Gaussian,

$$\tilde{f}(x, v, t) \sim F_M \sim e^{-v^2/2v_{th}^2}.$$

Since a Gaussian drops very quickly for large v , a Vlasov code can just take $\tilde{f}(x, v_{max}, t) = 0$ with no difficulty. By comparison, the PVT quantities, like $\tilde{n}^\dagger(x, v_p, t)$, fall asymptotically as $1/v_p$, which is much slower. This can be seen from the above boundary condition,

$$\begin{aligned}\tilde{n}^\dagger(x, v_p, t) &\sim Z_\pm \left[(v_p - V_0) / \sqrt{2} v_{th} \right] \\ &\longrightarrow \sqrt{2} v_{th} / (v_p - V_0) \quad (\text{large } v_p).\end{aligned}$$

It is fairly impossible to retain enough gridpoints so that f^\dagger becomes sufficiently close to zero. Possibly a good boundary condition can be found by assuming a smooth $1/v_p$ falloff of \tilde{n}^\dagger , but this has not been done yet.

B. semi-Eulerian calculation

The semi-Eulerian method begins with a shuffling around of the coordinates, boundary conditions, and fields, so that v_p goes from a coordinate to an Eulerian field, $v_p(x, t)$. The outcome of this transformation is that $\tilde{n}^\dagger(x, t)$, $\tilde{n}\tilde{V}^\dagger(x, t)$, $v_p(x, t)$ obey a set of Eulerian equations,

$$\partial_t \tilde{n}^\dagger + \partial_x (\tilde{n}^\dagger v_p) = 0 \quad (3.6)$$

$$\partial_t v_p + v_p \partial_x v_p = qE/m \quad (3.7)$$

$$\tilde{n}\tilde{V}^\dagger(x, t) = \tilde{n}^\dagger v_p \quad (3.8)$$

The boundary conditions for Eqs. (3.6-3.8) are similar to the Eulerian boundary conditions: \tilde{n}^\dagger and $\tilde{n}\tilde{V}^\dagger$ are initially satisfy Eqs. (3.3-3.4), for an initial array of v_p at each spatial point.

On the surface, it appears that the semi-Eulerian calculation is as computationally intensive as the full Eulerian calculation, since no approximation is made. But an important difference is that the semi-Eulerian formulation no longer has any v_p derivatives, ∂_{v_p} . This is important, because it means that it is no longer necessary to keep a continuous range of v_p with which to calculate derivatives. More specifically, only the v_p that resonate with the ϕ is, the new formulation of v_p allows a large simplification, which is that not all values of v_p must be retained. Therefore, in the semi-Eulerian scheme we can keep only the v_p that are resonant with the wave, and others can simply be dropped. For a monochromatic Langmuir wave, this means that we need a single v_p (and the concomitant \tilde{n}^\dagger and $\tilde{n}\tilde{V}^\dagger$) initialized to the phase velocity of the wave, which then accelerates in the wave potential as bouncing proceeds. Generally however, 1D Langmuir waves have *two* branches, one for each direction of propagation. Therefore, the semi-Eulerian calculation needs to retain *two* sets of phase velocity variables, one for each direction.

The semi-Eulerian closure method has been implemented in es1f with mixed results. It has been able to follow the nonlinear resonance dynamics of a damped Langmuir wave well, but only up to the point of the first “bounce,” which occurs when counter-streaming particles meet at the bottom of the potential well, and \tilde{n}^\dagger becomes double-valued. To rectify this, it is possible to numerically allocate two sheets for v_p and \tilde{n}^\dagger , to retain information on the two counter-streaming trapped species. [Fig. 1] However, the difficulty then becomes the boundary condition at the bottom of the well: initially, there is a steep jump in \tilde{n}^\dagger

at the bottom of the well that tends to diffuse away rapidly, due to numerical dissipation. [Fig. 2] These obstacles are probably surmountable with a little effort.

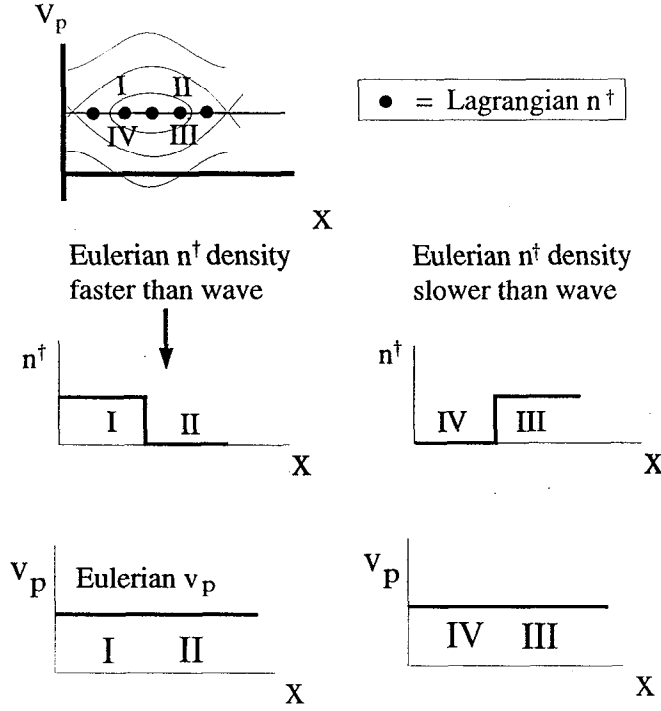


Figure 1: Illustration of the semi-Eulerian scheme, showing how two sheets of $\tilde{n}^\dagger(x, t_0)$ and $v_p(x, t_0)$ are initialized. [Next figure shows how evolution of trapping progresses.]

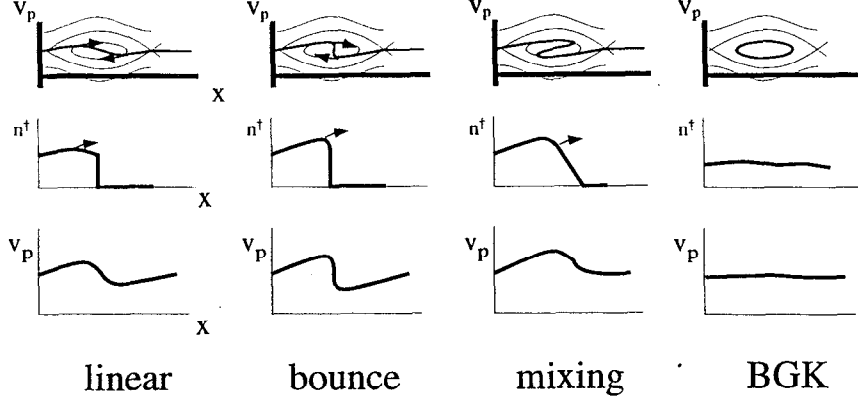


Figure 2: Semi-Eulerian method of handling trapping and phase mixing. One density field \tilde{n}^\dagger is retained for each side of the potential well; shown is one side of these. In the linear phase, the particles on one side of the potential well accelerate without crossing over. After a quarter-bounce time, they cross the middle of the well. After a half-bounce time, \tilde{n}^\dagger crosses to the other sheet of \tilde{n}^\dagger . The two species mix and form a BGK mode. Though the scheme is worked out conceptually, in practice it has proven difficult to find a differencing scheme that maintains the discontinuity in the linear phase and crosses over the bottom of the well at the right time in the bounce phase.

C. Local approximation

The local calculation is the simplest of all the nonlinear closures thus far attempted, and so far has proven the most successful. The equations follow from a method to approximate the PVT locally in space and time, described in MattorIV. The upshot of this is that \tilde{n}^\dagger gets a very simple approximation,

$$\tilde{n}^{\dagger s}(x, v_p, t) = \tilde{n}^s(x, t) \delta(v_p - v_p^{loc, s}), \quad (3.10)$$

where

$$\begin{aligned} v_p^{loc, s} &= -\frac{\partial_t \tilde{n}^s}{\partial_x \tilde{n}^s} \\ &= \frac{\tilde{n} \tilde{V}^s}{\tilde{n}^s} \end{aligned} \quad (3.11)$$

and,

$$\tilde{n}^s(x, t) \equiv \frac{1}{2}(1 - is\mathcal{H}_x)\tilde{n}^s(x, t)$$

with $s = \pm$ is the half of \tilde{n} with $sk > 0$. This approximation also gives $\tilde{n}\tilde{V}^\dagger$ as

$$\tilde{n}\tilde{V}^{\dagger s} = \tilde{n}^{\dagger s} v_p^{loc, s}.$$

Placing these terms into Eq. (5) gives a nicely simple expression for the pressure,

$$P \simeq -mnV^2 + T_0\tilde{n} + 2\text{Re} \left[v_{th} B_+ [(v_p^{loc, +} - V_0)/v_{th}] \tilde{n}\tilde{V}^+(x, t) \right]. \quad (3.12)$$

A nice feature of Eq. (3.11) is that it is fully local in time; that is, it can be calculated completely from any instantaneous fluid state, with no variables needed beyond the usual fluid fields $n(x)$ and $nV(x)$. [In contrast, the full Eulerian and semi-Eulerian methods require extra variables to retain the time histories.]

A difficulty which arises in the local method is “branch mixing,” defined as follows. For any given wavelength there are typically several branches, *i.e.*, multiple solutions for the phase velocity (or frequency). For the 1D electrostatic dynamics in eslf there are two branches: left and right traveling Langmuir waves. A general \tilde{n} and \tilde{nV} are superpositions of these waves, which might be written as

$$\begin{aligned}\tilde{n} &= \sum_i \tilde{n}_i \\ \tilde{nV} &= \sum_i \tilde{nV}_i,\end{aligned}$$

where each wave i has its own phase velocity and resonance dynamics. On the other hand, the local phase velocity as given by Eq. (3.11) does not separate waves, so produces some hybrid v_p that does not represent any actual wave. This can lead to numerical difficulties like inaccuracy or instability. It is necessary to separate fluctuations the various branches, and calculate the closure part of the pressure separately for each i , with phase velocity

$$v_{p,i}^{loc,s} = \frac{\tilde{nV}^s}{\tilde{n}^s}. \quad (3.13)$$

This is possible because the Vlasov equation and the closure term are linear in f , hence the closure term is superposable.

Separating \tilde{n} and \tilde{nV} into the separate branches [here left and right propagating waves], has been surprisingly difficult. The most successful method so far has been to

- (1) Complexify the evolution equations,
- (2) Write the evolution equations in eikonal form, as a polynomial in v_p , then,
- (3) Factor them into components.

The case of a single species electron fluid allows a simple illustration. Two fields can be eliminated via $\tilde{n} = -\partial_x \tilde{E}$ and $\tilde{n} = \partial_t \tilde{E}$, and the equation complexified, leaving

$$[\partial_t^2 - v_{th}^2 \partial_x^2 + \omega_{pe}^2] \tilde{E}^+ = \tilde{nE}^+ - B(v_p/v_{th}) \partial_x \partial_t \tilde{E}^+. \quad (3.13)$$

Assuming the usual eikonal ordering for Eq. (3.13) allows $\partial_t \rightarrow -i\omega$ and $\partial_x \rightarrow -ik$ and produces

$$\left[\omega^2 - B(v_p/v_{th})k\omega - v_{th}^2 k^2 - \omega_{pe}^2 - \tilde{nE}^+/\tilde{E}^+ \right] \tilde{E}^+ = 0.$$

This can be factored to produce,

$$[v_p - v_{p\pm}^{loc}] [v_p \tilde{E}^+ - v_{p\mp}^{loc} \tilde{E}^+] = 0, \quad (3.14)$$

where

$$v_{p\pm}^{loc} \equiv B/2 \pm \sqrt{B^2/4 + 1 + (1 + \tilde{nE}^+/\tilde{E}^+)/k^2},$$

where $k = i\tilde{n}^+/\tilde{E}^+$ in the eikonal approximation. Written this way, Eq. (3.14) shows that the term in the second brackets is a wave, and the operator in the first brackets makes it vanish. Therefore, we can write the two waves \tilde{E}_\pm^+ as

$$\tilde{E}_\pm^+ = [v_p^{loc} - v_{p\mp}^{loc}] \tilde{E}^+ \quad (3.15)$$

$$v_{p\pm} = v_{p\pm}^{loc}, \quad (3.16)$$

where $v_p^{loc} = \tilde{n}\tilde{V}^+/\tilde{n}^+$ as above. Separating the wave this way and calculating two pressure terms appears to fix a numerical instability associated with excitation of a back-propagating wave.

Separating the wave this way works well for the 1-species O'Neil problem, but difficulties arise when generalizing it to multiple species. First, the size of the matrix to be solved increases like twice the number of species, which becomes rapidly cumbersome. Second, it does not allow for waves which may be introduced from a source outside the system. Third, it incorporates coupling between different fluid species, which is cumbersome and also unnecessary in a collisionless system: the only coupling is from a given species to the electric field. There is a better method which circumvents all these difficulties. First one should somehow divide the wave in ϕ into separate waves. Then, for each fluid species calculate the waves that arise in response to the waves in the field, which is independent of the other species. The linearity of the Vlasov equation in f should be of great help here. For each wave in ϕ there should be two waves in each species, one for the "forward" wave (which uses the positive branch of the Z_s function) and one for the "backward" wave. We are currently working on methods to do this.

D. Linearized calculation (Chang-Callen)

Early work by Ott and Sudan and later by Chang and Callen calculated collisionless closure for *linearized* equations, which is significantly more restrictive than the nonlinear closure above. However, we can always linearize the closure term, which facilitates calculation. The result is similar to the local approximation above, except that $v_p^{loc,+}$ is replaced by the linear phase velocity, v_p^{lin} , calculated as an eigenvalue of the linear dispersion relation.

$$P \simeq -mnV^2 + T_0\tilde{n} + 2\text{Re} \left[v_{th}B_+[(v_p^{lin} - V_0)/v_{th}]\tilde{n}\tilde{V}^+ \right]. \quad (3.17)$$

It is relatively straightforward to calculate this term, and no difficulty seems to arise in the code runs [like the branch mixing mentioned above]. The difficulty with the linearized closure is that it misses nonlinear effects such as particle trapping, since v_p^{lin} is constrained to being a single value, without feedback from the nonlinearly evolving wave. Thus linearizing the CCT simplifies computation quite a bit, but sacrifices much of the nonlinear trapping the dynamics. In particular, the Landau pole is always fixed on a single branch, and is unable to display the dynamical reversal when bouncing occurs [MatterI, MatterII].

E. Linearized $v_p = 0$ calculation (Hammet-Perkins)

Following the work of Hammet and Perkins, it is possible to simplify further beyond linearization, and take the limit where the argument of B is zero. This gives a great simplification to the closure,

$$P \simeq -mnV^2 + T_0\tilde{n} - v_{th}\sqrt{\pi/2}\mathcal{H}_x \left(\tilde{n}\tilde{V} \right). \quad (3.18)$$

This has even further simplicity, since there is no v_p that needs to be calculated, but has the same loss of bounce dynamics that the linearized calculation has, and a further loss of accuracy.

IV. Problems addressed

The calculations mentioned in the previous section were developed while considering two basic problems, which we call “The O’Neil Problem” and “Two-Stream Instability.” Work on these is still progressing and the ideal solution has yet to be found, but in this section we discuss some of the results and obstacles encountered.

A. The O’Neil Problem

“The O’Neil problem” refers to the situation addressed by O’Neil in 1965 [O’Neil]. A Maxwellian 1D electrostatic plasma is initialized with a Langmuir wave. Initially the resonant particles accelerate in the wave potential, and the wave undergoes Landau damping. But with time, the resonant particles travel far enough to cross the potential well bottom, and then begin to decelerate. Since energy is conserved, the Landau damping reverses and the wave grows. The particles bounce back and forth like this and eventually mix and become a “BGK mode.” [Sagdeev and Galeev].

A large part of this project was spent considering this problem, and much progress was made. All of the above methods except the full Eulerian calculation were able to follow the linear damping phase with suitable accuracy. (Fig. 3) The nonlinear phase proved considerably trickier however, for the unexpected reason that *the benchmark code, es1, gives ambiguous results*. More specifically, as seen from Fig. 3, es1 damps at the linear rate for a couple orders of magnitude, then the damping ceases. This would appear to be from trapping, but according to [Denavit and Walsh 1981] the time of saturation is sufficiently shorter than a bounce time to rule out any trapping effect. Denavit and Walsh also cite a kinetic Vlasov code, which show no trapping at all, and conclude that the apparent bouncing in es1 comes from an insufficient number of particles in the vicinity of the resonance. By the time the wave has damped to its former bounce level, the number of resonant particles is even less. The upshot of this is that at high wavenumber, where Landau damping is strong, es1 shows spurious bouncing. At low wavenumber the Landau damping is too weak to be reliably diagnosed. Thus, we had no reliable way of telling if es1f is accurate, and so this was basically not a good problem to carry to completion.

Nonetheless, a wealth of information was learned from the O’Neil problem, including the development of the plethora of computational schemes discussed above. The good news, mostly expected, was that es1f outperforms es1 in several ways. The results were more accurate, with no appreciable noise and no spurious bouncing. The fluid code was much faster, since it used a 1D spatial grid, not the 2D grid (x, v) . Finally, es1f unexpectedly requires far fewer spatial grid points than es1: for the results shown, es1f required 32 grid points to obtain results comparable to es1 runs with 256 grid points. The reason for this large savings is not known.

Wave Energy from different codes

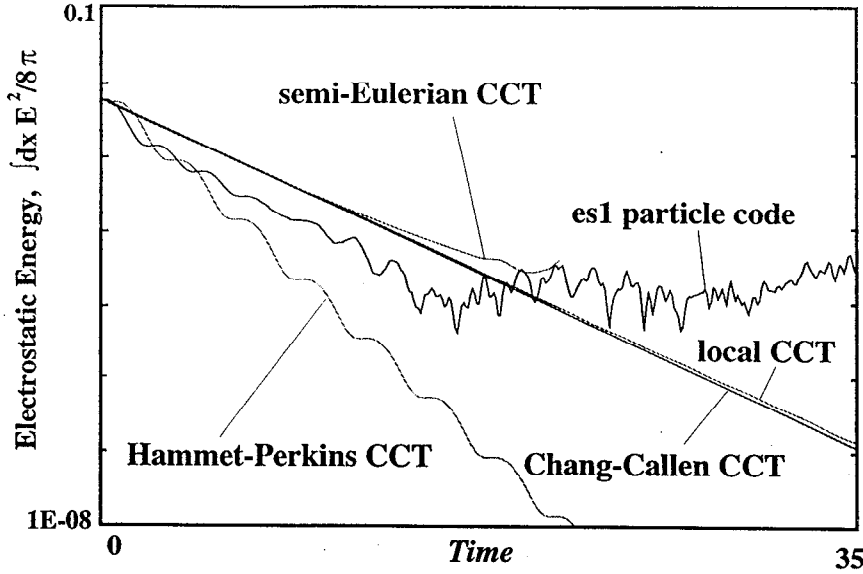


Figure 3: Comparison electrostatic energy history from several computational schemes. The codes were started with $k\lambda_D = 0.5$ and $\tilde{n}/n_0 = 0.02$. The wave decayed by Landau damping.

- **es1:** Energy decayed at first by the linear decay rate. The decay stopped after energy diminished by about 2 orders of magnitude. According to Denavit and Walsh, the saturation of the decay is earlier than a bounce time, and therefore spurious.

- **semi-Eulerian CCT:** The semi-Eulerian method described above initially damps at the linear rate. As the resonant particles fall toward the bottom of the potential well the damping begins to stop. The code crashes at that point, because the particles meeting at the bottom of the well are constrained to give single-valued density \tilde{n}^\dagger [Fig. 4]. The code needs to be modified to allow convergent particles to move around each other.

- **local CCT:** Energy damped steadily at the linear rate. According to [Denavit and Walsh, 1981], this is the correct result in this range, since it agrees with a kinetic Vlasov code.

- **Chang-Callen CCT:** The damping is linear, and largely indistinguishable from the local CCT result.

- **Hammet-Perkins CCT:** Energy damps steadily at about twice the correct rate. The oscillations come from the initial wave having a benign component of standing wave.

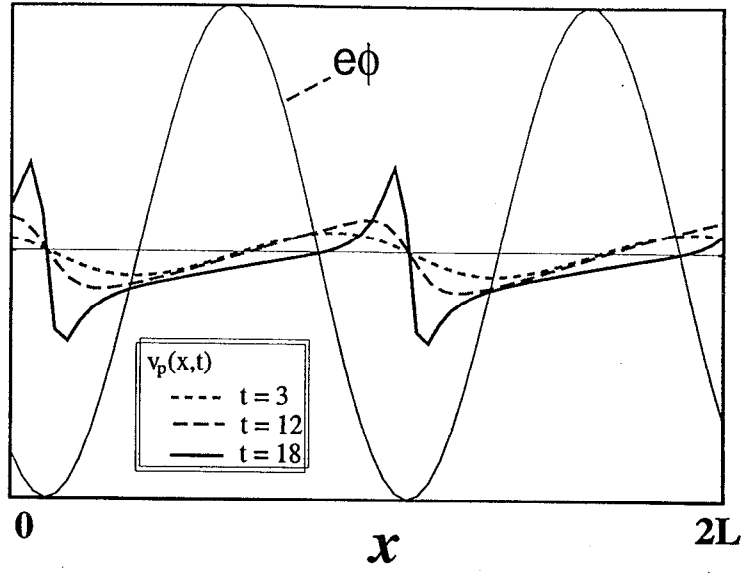


Figure 4: Evolution of the semi-Eulerian phase velocity relative to the potential, in the rest frame of the wave. The velocity falls toward the center of the potential well until opposing streams meet at the bottom. The code is not yet designed to evolve beyond this phase.

B. Two-Stream Instability

After realizing the unsuitability of the O'Neil problem for nonlinear testing of es1f, we turned attention to the Two-Stream Instability. This is another classic problem, where two counter-streaming electron beams are electrostatically unstable [Birdsall and Langdon]. The unstable mode grows up to nonlinear levels, and when the accelerated electrons cross the bottom of the potential well the wave begins to damp. Resonant electrons continue to orbit back and forth, phase mixing ensues, and the instability ends up as a steady BGK mode which lasts a long time. The fact that bouncing occurs when the wave is at its largest amplitude, not smallest, makes this better than the O'Neil problem for numerically studying nonlinear trapping dynamics, phase mixing, and asymptotic formation of the BGK mode.

The local approximation described above has been able to follow the two-stream instability very accurately up to the time of the first bounce. In several important ways, es1f has proven superior to es1. There is no detectable particle noise, and it is far faster and more efficient because it saves a dimension. Figs. 5 and 6 show the 2-stream instability as described by es1 and es1f, respectively. The agreement is quite good up the time of the first bounce.

The current problem that needs to be surmounted in es1f is the difficulty that arises at the first particle bounce. What happens is that the particles which have bounced collide with the particles that haven't. In the kinetic system, the extra dimension allows the particles to travel *around* each other, but in the fluid system there is no such route. The over-compression causes numerical instability and es1f crashes. The density is trying to become double valued, as it does with a breaking wave, but es1f does not currently allow this.

What is needed to surmount this problem is (1) to develop some criterion for recognizing such a bounce, and (2) divide the bounced part of the fluid from the unbounced part, and (3) reassign the bounced part to another sheet or to another species. The "unbounced" and "bounced" parts of the species, with forward and reverse Landau damping, respectively, which are to be assigned the $+$ and $-$ branches of the Z_{\pm} -function. The separated part of the bouncing particles can be reassigned in several places, such as to a second sheet of \tilde{n}^{\dagger} [such as we have done in the semi-Eulerian computation above]. Another possibility is to create a species of "clumps" or "holes" that could coexist with the fluid plasma, which are launched at the time of a bounce. We are currently working on methods to do steps (1) and (2) above, such as the factoring of the locally approximated dynamical equations described in the "local approximation" section above.

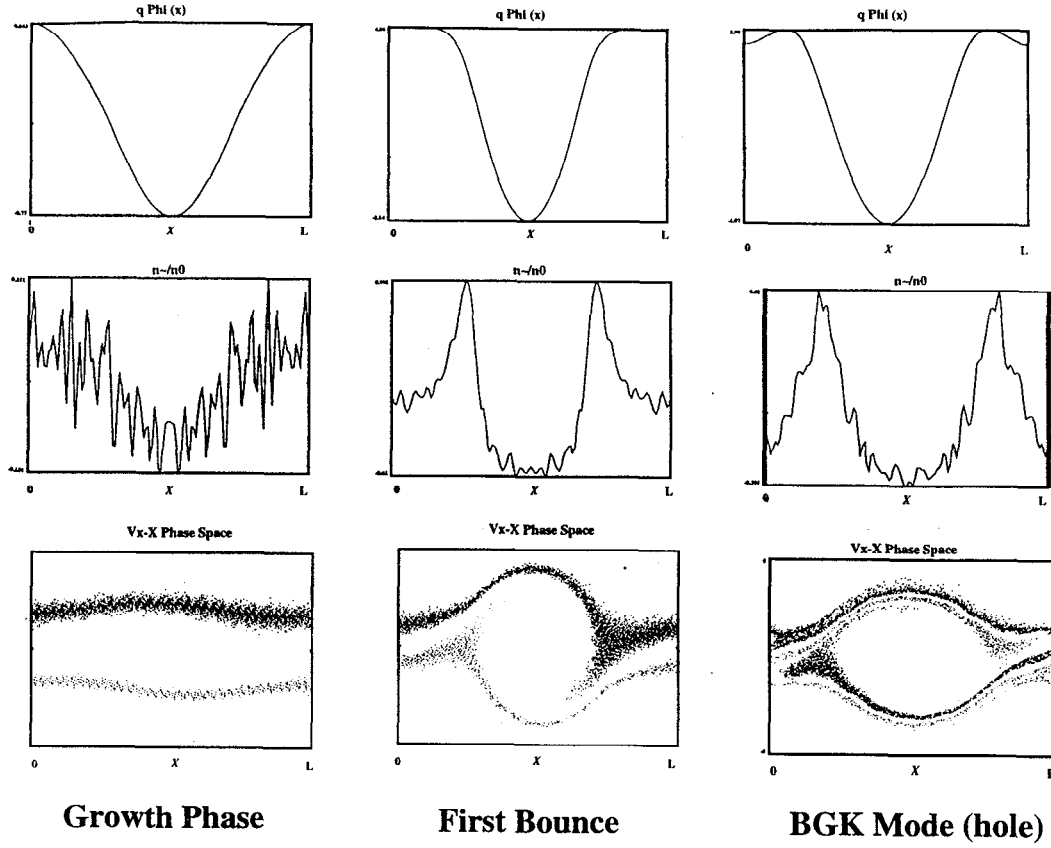


Figure 5: Kinetic picture of 2-stream instability, as given by es1. In the instability phase, a sinusoidal wave grows. In the bounce phase, the trapped particles bounce in the electrostatic well. After many bounces, the trapped particles are phase mixed into a BGK mode, also known as a “hole.”

Notice that the density is very rough, due to noise from particle discreteness.

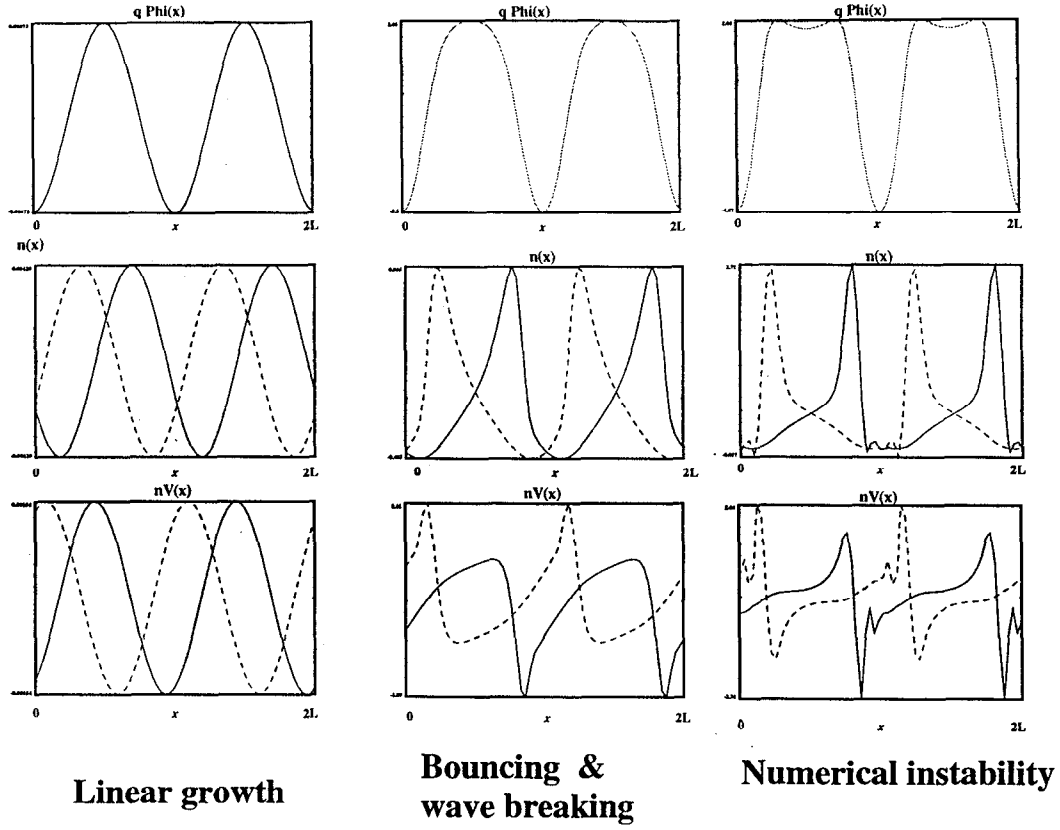


Figure 6: Fluid picture of 2-stream instability, as given by eslf with the “local” approximation to the CCT. The solution is quite faithful to the kinetic solution, up to the point of the first bounce, where the eslf is not yet designed to handle the multi-valued density that ensues. Interestingly that the numerical instability is a grid-scale oscillation in nV that occurs *in the region where the kinetic code has split the species in two*; it is almost as if the fluid code is trying to obey this double-valuedness as best it can.

This fluid code runs much faster than the kinetic code, and the particle noise in \tilde{n} is clearly much reduced.

V. Hamiltonian Collisionless Closure Term

An important accomplishment of this project was generalizing the CCT from the 1D electrostatic Vlasov equation, Eq. (2.1), to a much more general “Hamiltonian Vlasov equation”

$$\partial_t f + (\partial_{p_i} H) \partial_{q_i} f - (\partial_{q_i} H) \partial_{p_i} f = 0, \quad (5.1)$$

where (q_i, p_i) are the i^{th} components of canonical position and momentum coordinates. For later convenience, we assume the Hamiltonian to have the quadratic form

$$H(\mathbf{q}, \mathbf{p}, t) = \frac{1}{2}(\mathbf{p} - \mathbf{a}) \cdot \mathbf{W} \cdot (\mathbf{p} - \mathbf{a}) + U$$

where $\mathbf{W}(\mathbf{q}, t)$, $\mathbf{a}(\mathbf{q}, t)$, $U(\mathbf{q}, t)$ and the dimension D are arbitrary. Such a Hamiltonian encompasses many collisionless systems, including an ideal gas ($H = p^2/2m$), a self-gravitating astrophysical gas ($H = p^2/2m + m\psi$) a plasma in an electro-magnetic field ($H = (\mathbf{p} - e\mathbf{A}/c)^2/2m + e\phi$), or even a Liouville equation for N particles ($H =$ some function on $6N$ dimensional space). Thus, the CCT for this equation has a very wide range of applications.

In a yet-to-be-released paper, the collisionless closure procedure has been applied to Eq. (5.1). Here, we skip the arduous details, and just state the results. For compactness, the moment equations are truncated at the first, giving

$$\partial_t \rho + \partial_{q_i} (\rho V_i) = 0 \quad (5.2)$$

$$\partial_t \rho V_i + \partial_{q_j} [\rho V_i V_j] = \mathcal{F}_i - \partial_{q_j} P_{ij}, \quad (5.3)$$

where

$$\begin{aligned} \rho &= \int d^D p \\ \rho V_i &= \int d^D p (\partial_{p_i} H) f \\ \mathcal{F}_i &= \int d^D p (\partial_t \partial_{p_i} H + \{H, \partial_{p_i} H\}) f \\ P_{ij} &= \int d^D p (\partial_{p_i} H - V_i)(\partial_{p_j} H - V_j) f \end{aligned}$$

As before, the pressure term requires closure. After a great deal of calculation, it comes out to

$$\begin{aligned} P_{ij}(\mathbf{q}, t) &= \int dv_p \sum_{s=\pm} \mathcal{H}_{\mathbf{q}_{\parallel}} (1 - is\mathcal{H}_{\mathbf{q}_{\parallel}}) \left[\frac{\rho_0 |Det(\mathbf{W})|}{(2\pi T_0)^{D/2}} \right]_{\mathbf{q}_p(\mathbf{q}, \mathbf{v}_p; t_0)} \mathcal{Z}_s(\mathbf{q}, \mathbf{v}_p, t) \\ \mathcal{Z}_{\pm}(\mathbf{q}, \mathbf{v}_p, t) &= \int d^D p \frac{[\mathbf{W}(\mathbf{p} - \delta\mathbf{p}) - \mathbf{V} - \mathbf{V}_{0p}]_i [\mathbf{W}(\mathbf{p} - \delta\mathbf{p}) - \mathbf{V} - \mathbf{V}_{0p}]_j}{|\mathbf{W}(\mathbf{p} - \delta\mathbf{p}) - \mathbf{V}_{0p}| - v_p^{\pm}} \\ &\quad \times \exp \left[-\frac{1}{2T_{0p}} \mathbf{p} \cdot \overline{\mathbf{W}} \cdot \mathbf{p} \right] \end{aligned}$$

$$\begin{aligned}
\overline{\mathbf{W}} &= [1 + \mathbf{A}]^T \mathbf{W} [1 + \mathbf{A}] \\
\delta \mathbf{p} &= [1 + \mathbf{A}]^{-1} \left\{ \int_t^{t_0} dt' \mathbf{f}(\mathbf{q}_p(\mathbf{q}, \mathbf{v}_p, t'), t') - \mathbf{A} \mathbf{v}_p \right\} \\
\mathbf{f} &= -\nabla_{\mathbf{q}} U - \partial_{t'} \mathbf{a} + \mathbf{v}_{p,i} \nabla_{\mathbf{q}} a_i - \nabla_{\mathbf{q}} \mathbf{a} \cdot \mathbf{v}_p \\
\mathbf{A} &= \int_t^{t_0} dt' \nabla_{\mathbf{q}_p} \mathbf{a}(\mathbf{q}_p, t')
\end{aligned}$$

While this work has the potential to be extremely valuable, I have come to the realization that no one will have the attention span to follow it unless I can make the simpler version more accessible. [For example, I seriously doubt if anyone will ever read these words I am writing now.] Therefore, I decided that before releasing this generalized version, it is important to make the simplified version [1D electrostatic CCT] able to be computed in a simple code that requires very little brainpower. For this reason, the generalized Hamiltonian work above was put on the back burner and we returned to writing eslf.

VI. Phase Velocity Transform

The most dramatic and useful result of the last year has been a “spin-off” project, which uses the mathematics developed for deriving the CCT. Specifically, in deriving the CCT, we developed the “Phase Velocity Transform” (PVT), which is an integral transform that divides a function of space and time into components propagating at constant phase velocities without distortion [Fig. 7].

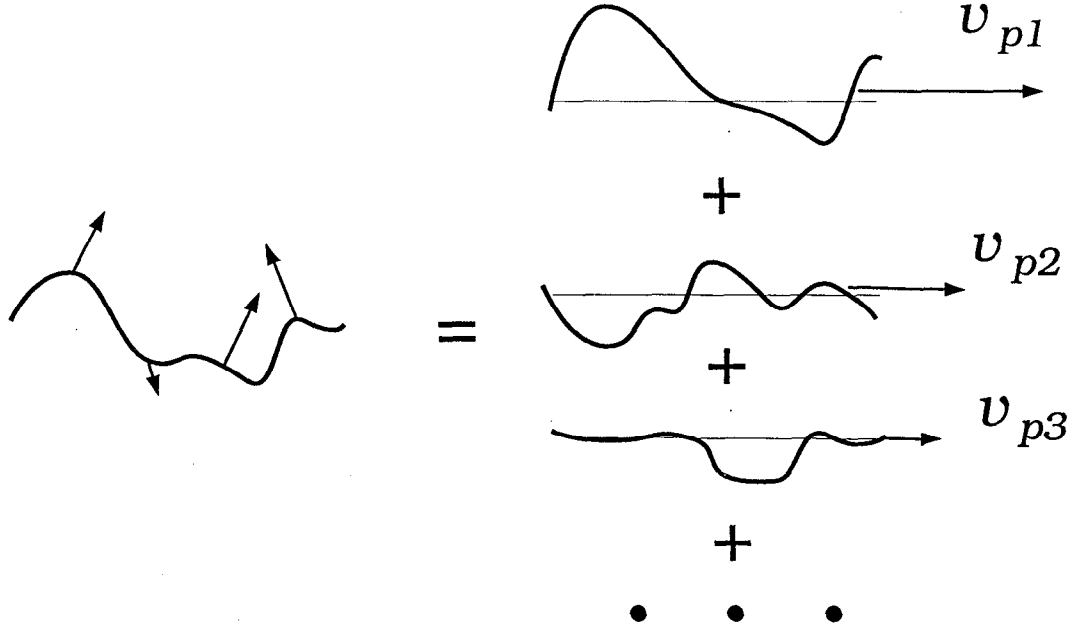


Figure 7: Cartoon of phase velocity transform, which divides a function of space and time, $f(x, t)$, into components which propagate at constant phase velocities, without distorting.

Though originally developed for solving differential equations, the PVT also has a use in decomposing spatio-temporal fluctuations. It has at least two advantages over other

methods currently in common use (*e.g.*, Fourier analysis, singular value decomposition, and wavelets).

First, it avoids imposing basis functions with some preset form, such as sine waves or wavelets. Second, the PVT does not force space-time separation of basis functions, $f(x, t) = \sum A_i(t)\hat{u}_i(x)$, which can be awkward in representing propagating waves, which generally have an inseparable space-time dependence $u_i(x - v_p t)$. For both these reasons, the PVT is good for picking propagating structures of unknown shape out of noisy backgrounds. These occur in a number of physical systems, particularly nonlinear fluids,

To develop this the spatio-temporal signal analysis aspect, the PVT was used to decompose fluctuation data from the Sustained Spheromak Physics Project (SSPX) at LLNL, as an alternative to standard techniques such as Fourier or wavelet transforms, or singular-value decomposition. In fact, we have learned very much about the SSPX fluctuations that would not be otherwise available from the various other transforms, regarding non-sinusoidal shapes of the waves, nonlinear interactions of the waves, and a more detailed view of the spectrum than is otherwise available. More information can be found in UCRL-JC-136756 [MattorV], which has been submitted to Physical Review Letters.

References

1. Birdsall and Langdon, *Plasma Physics via Computer Simulation*, (copyright 1985, McGraw-Hill, New York).
2. Chang, Z., and J. D. Callen, *Phys. Fluids B* **4**, 1167 (1992).
3. Denavit, J., and J. M. Walsh, *Nonrandom initializations of particle codes*, *Comments Plasma Phys. Control. Fusion* **6**, 209-223 (1981).
4. Hammett, G. W., and F. W. Perkins, *Phys. Rev. Lett.* **64**, 3019 (1990).
5. Mattor, N., and S. E. Parker, *Nonlinear kinetic-fluid equations*, *Phys. Rev. Lett.* **79**, 3419 (1997). MattorI
6. Mattor, N., *Collisionless fluid closure theory and the phase velocity transform*, *Physics of Plasmas* **5**, 1892 (1998). MattorII
7. Mattor, N., *Local Approximation for the Phase Velocity Transform*, *Physics of Plasmas* **6**, 1059 (1999). MattorIII
8. Mattor, N., *Collisionless fluid equations: Lagrangian, Eulerian, and local forms*, *Physics of Plasmas* **6**, 1065 (1999). MattorIV
9. Mattor, N., *Spatiotemporal Signal Analysis via the Phase Velocity Transform in SSPX* (to be submitted *Phys. Plasmas*). MattorV
10. Mattor, N., *Spatiotemporal Signal Analysis via the Phase Velocity Transform* UCRL-JC-136756 (sub. *Phys. Rev. Lett.*). MattorVI
11. O'Neil, T. M. *Phys. Fluids* **8**, 2255 (1965).
12. Ott E., and R. N. Sudan, *Phys. Fluids* **12**, 2388 (1969).
13. Sagdeev, R. Z., and A. A. Galeev, *Nonlinear Plasma Theory*, edited by T. M. O'Neil and D. L. Book (Benjamin, New York, 1969).